AIAA JOURNAL
Vol. 39, No. 5, May 2001

Survey of Shape Parameterization Techniques for High-Fidelity Multidisciplinary Shape Optimization

Jamshid A. Samareh*
NASA Langley Research Center, Hampton, Virginia 23681

A survey is provided of shape parameterization techniques for multidisciplinary optimization, and some emerging ideas are highlighted. The survey focuses on the suitability of available techniques for multidisciplinary applications of complex configurations using high-fidelity analysis tools such as computational fluid dynamics and computational structural mechanics. The suitability criteria are based on the efficiency, effectiveness, ease of implementation, and availability of analytical sensitivities for geometry and grids. A section on sensitivity analysis, grid regeneration, and grid deformation techniques is also provided.

Nomenclature

B = Bernstein polynomial
$\bar{c}$ = polynomial coefficients
$\bar{D}$ = grid perturbations
J = cell Jacobian
k = spring stiffness
N = B-spline basis function
$\bar{P}$ = coordinates of nonuniform rational B-spline (NURBS) control point
$\bar{R}$ = coordinates of deformed model
$\bar{r}$ = coordinates of baseline model
t = response
$\bar{U}$ = design vector
u = independent parameter coordinate
V = baseline cell volume
$\bar{v}$ = design variable vector
W = NURBS weights
ε = small positive number

Subscripts

f = field (volume) grid
g = geometry
i, j = control point indices
k = grid-point index
m = element index
n = basis vector index
p = degree of Bernstein polynomial and B-spline basis function
s = surface grid

Superscripts

i = polynomial power
n = number of design variables

Introduction

IMAGINE that you have been asked to perform multidisciplinary shape optimization (MSO) for a complete aerospace configuration during the design phase. In MSO, the use of high-fidelity tools such as computational fluid dynamics (CFD) and computational structural mechanics (CSM) will bring more confidence to the design. These tools require detailed grid models. The availability (or lack) of automated commercial grid generation tools plays an important role in the selection of a shape parameterization approach.

Generally, multidisciplinary design optimization (MDO) should exploit the synergism of the primary, mutually interacting phenomena to improve the design. The MDO applications commonly involve sizing, topology, and shape optimization. Sizing optimization is used to find the optimum cross-sectional area for bars and trusses and thickness for plate and shell elements. Sizing optimization is a matured technology and is available in most commercial CSM tools. Topology optimization is a technique for determining the optimal material distribution, which can suggest the optimum layout of the structure. Shape optimization finds the optimum shape for a given structural layout. Obviously, the choice of shape parameterization technique has enormous impact on the formulation and implementation of the optimization problem. This paper reviews and evaluates the available shape parameterization techniques for multidisciplinary optimization of aerospace applications.

Over the past several decades, single discipline shape optimization has been successfully applied to two-dimensional and simple three-dimensional configurations.[1,2] In recent years, interest in the application of MSO to complex three-dimensional configurations has grown.[3] The MSO for a complete aerospace configuration is a challenging task, especially if the MSO application is based on high-fidelity analysis tools. This survey focuses primarily on shape parameterization for high-fidelity analysis tools such as CFD and CSM because geometry and grid generation requirements for these tools are the most stringent among the disciplines used in the MDO of an aerospace vehicle.

CFD tools use the detailed definition of the skin shape (also referred to as the outer mold line), whereas CSM tools use all components. Generally, the CSM models require only a relatively coarse grid, but the grid must handle very complex internal and external geometry components. In contrast, the CFD field grid is very fine, but only needs to model the external geometry components. The MSO of an aerospace vehicle must treat not only the external geometry, for example, wing skin, fuselage, flaps, nacelles, and pylons, but also the internal structural elements, for example, spars, stiffeners, ribs, and fuel tanks, as shown in Fig. 1.

Sensitivity analysis is an important consideration for shape parameterization, and the next section provides details. Then, several shape parameterization approaches are reviewed. The last section of the paper provides an overview of grid deformation and grid regeneration methods.

Sensitivity Analysis

Sensitivity is defined as the partial derivative of a response with respect to a design variable. The sensitivity analysis is an essential building block of gradient-based optimization. In some of the CSM literature, the sensitivity derivatives are referred to as the design velocity field. Despite recent advances in sensitivity analysis, very

Presented at the CEAS/AIAA/ICASE/NASA Langley International Forum on Aeroelasticity and Structural Dynamics, Williamsburg, VA, 22–25 June 1999; received 1 November 1999; revision received 22 September 2000; accepted for publication 17 October 2000.

*Senior Research Scientist, Multidisciplinary Optimization Branch. Senior Member AIAA.

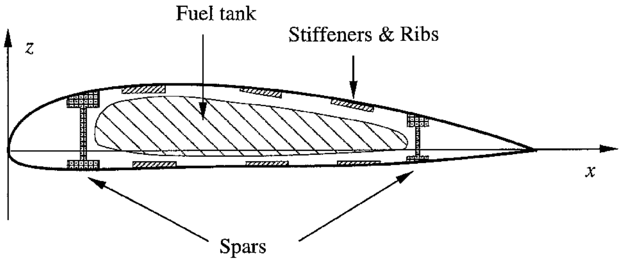


Fig. 1 Internal components of a wing.

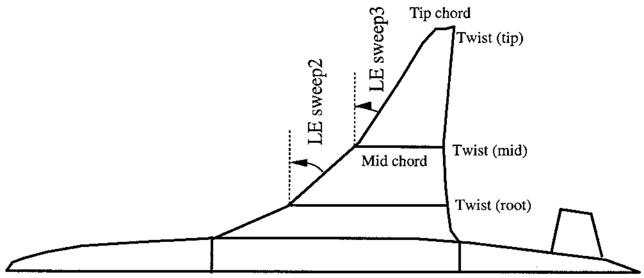


Fig. 2 Design variables for a high-speed civil transport.

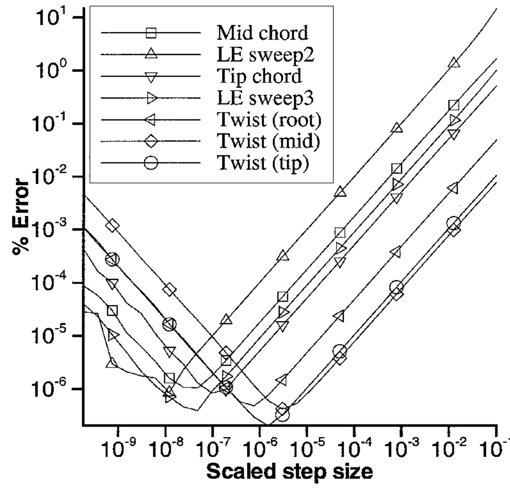


Fig. 3 Error in finite difference approximation for shape sensitivity derivative calculations.

few geometry modeling and grid generation tools currently provide analytical sensitivity.

The sensitivity derivatives of a response t with respect to the design variable vector $\bar{v}$ can be written as

$$\frac{\partial t}{\partial \bar{v}} = \frac{\partial t}{\partial \bar{\boldsymbol{R}}_f}\frac{\partial \bar{\boldsymbol{R}}_f}{\partial \bar{\boldsymbol{R}}_s}\frac{\partial \bar{\boldsymbol{R}}_s}{\partial \bar{\boldsymbol{R}}_g}\frac{\partial \bar{\boldsymbol{R}}_g}{\partial \bar{v}} \tag{1}$$

where $\bar{\boldsymbol{R}}_f$ is the field (volume) grid, $\bar{\boldsymbol{R}}_s$ is the surface grid, and $\bar{\boldsymbol{R}}_g$ is the geometry description.

The first term on the right-hand side of Eq. (1) represents the sensitivity derivatives of the response with respect to the field grid-point coordinates. For a detailed discussion, see Refs. 1–4 for CSM discipline and Refs. 3 and 5 for CFD discipline. The second term on the right-hand side of Eq. (1) is vector of the field grid-point sensitivity derivatives with respect to the surface grid points. The field grid generator must provide the sensitivity derivative vector, but few grid generation tools have the capability to provide the analytical grid-point sensitivity derivatives.[6] The third term on the right-hand side of Eq. (1) denotes the surface grid sensitivity derivatives with respect to the geometry, which the surface grid generation tools must provide these derivatives. The fourth term on the right-hand side of Eq. (1) signifies the geometry sensitivity derivatives with respect to the design variable vectors; the geometry construction tools, such as CAD, must provide this term.

Zang and Green have reviewed current methods and tools for sensitivity analysis.[7] There are four techniques for computing the sensitivity derivatives: manual differentiation, automatic differentiation, complex variables, and finite difference approximation. If the source codes are available, they can be differentiated either by hand or with automatic differentiation tools.

Automatic differentiation tools such as ADIFOR[8] or ADIC[9] can simplify and automate the differentiation process. Argonne National Laboratory maintains a web site on computational differentiation tools[†] such as ADIC[‡] and ADIFOR.[§] These are preprocessing tools. For example, ADIFOR accepts as input a FORTRAN code, along with specifications of the input and output variables. ADIFOR then produces an augmented FORTRAN code that contains the original analysis capability plus the capability for computing the analytical derivatives of all of the specified output quantities with respect to all of the specified input quantities. Another attractive alternative is the use of the complex variable technique.[10,11] Of course, a hand-coded differentiation will probably be more efficient in terms of both computation time and computer memory.[7]

It is possible to use finite differences to approximate the sensitivity derivatives, but feasibility and accuracy issues must be considered. Using finite difference approximations for sensitivity calculation is feasible as long as the perturbed geometry (grids) has the same topology as the unperturbed geometry (grids). Commercial CAD systems and unstructured grid generation techniques do not guarantee the same topology for the perturbed and unperturbed geometries (grids).

Figure 2 shows a high-speed civil transport with seven planform design variables. Figure 3 shows the error involved in using a finite difference approximation for shape sensitivity derivative calculations [fourth right-hand term in Eq. (1)]. This error behavior is typical of finite difference approximations for sensitivity calculations. For larger step sizes, the truncation error is predominant, and for smaller step sizes, the roundoff error is predominant. At the optimal step size, the error is minimum. This optimal step size is different for each design variable and would also vary for each optimization cycle. Using higher-order approximation and techniques outlined in Refs. 12 and 13 can reduce the truncation error, provided that geometry (grid) topology stays fixed during the evaluation of the finite difference approximation.

Multidisciplinary Shape Parameterization

Eight shape parameterization approaches are reviewed in this section; the discussion focuses on the suitability of available techniques for multidisciplinary high-fidelity applications of complex configurations. A successful parameterization process must 1) be automated, 2) provide consistent geometry changes across all disciplines, 3) provide sensitivity derivatives (preferably analytical), 4) fit into the product development cycle times, 5) have a direct connection to the CAD systems used for design, and 6) produce a compact and effective set of design variables for the solution time to be feasible. For more details, see Ref. 14.

The high-fidelity analyses involved in an MDO application normally use grids, and these disciplinary grids are often dissimilar. An MDO application demands consistent shape changes across all disciplines. Either parameterizing all disciplinary grids consistently or using a single source for geometry creation can achieve consistent shape parameterization. Manual creation of CFD and CSM grids is time consuming and costly for a full airplane model: It takes several months to develop detailed CSM and CFD grids based on a CAD model. The disciplinary grids can be parameterized, and then they can be deformed. The grid deformation alleviates the need for automatic grid generation tools, but its use is limited to small geometry perturbations.[14] On the other hand, the single source for geometry is very attractive, but requires the grid be regenerated during the optimization process. This requirement compels the MDO environment to rely on automatic grid generation tools, which may not be available for all disciplines.

During the design optimization, the geometry goes through many small geometry perturbations. For example, the optimization of a

[†]See URL http://www-unix.mcs.anl.gov/autodiff.
[‡]See URL http://www-fp.mcs.anl.gov/adic.
[§]See URL http://www-unix.mcs.anl.gov/autodiff/ADIFOR.

wing starts with a baseline wing design, and the goal is to improve the wing performance by using numerical optimization. The geometry changes (perturbations) between the initial and optimized wings are very small,[15,16] but the difference in wing performance can be substantial. When the shape perturbations are parameterized instead of the shape itself, the number of shape design variables can be reduced, and the need to reverse-engineer existing nonparametric models can be avoided entirely.

An important ingredient of shape optimization of an aerospace vehicle is the availability of a model parameterized with respect to the airplane shape parameters such as planform, twist, shear, camber, and thickness.

The parameterization techniques can be divided into eight categories: basis vector, domain element, partial differential equation, discrete, polynomial and spline, CAD-based, analytical, and free-form deformation (FFD). Haftka and Grandhi[1] and Ding[2] provide surveys of shape optimization up to 1986. The present focus is on some recent developments in the area of shape parameterization for complex aerospace models and their suitability for MSO applications. The suitability criteria are based on the efficiency, effectiveness, ease of implementation, and availability of analytical sensitivities for geometry and grid models.

Basis Vector Approach

Pickett et al.[17] proposed a technique that combines the second through fourth right-hand terms of Eq. (1) into a set of basis vectors. The shape changes can be expressed as

$$\bar{\boldsymbol{R}} = \bar{\boldsymbol{r}} + \sum_{n} \bar{v}_n \bar{\boldsymbol{U}}_n \tag{2}$$

where $\bar{\boldsymbol{R}}$ is the design shape, $\bar{\boldsymbol{r}}$ is the baseline shape, and $\bar{\boldsymbol{U}}$ are design vectors based on several proposed shapes. The proposed shapes must share the same grid topology. When the shape changes are parameterized, this technique can provide a compact set of design variables. With the assumption that the reduced basis is constant throughout the optimization cycle, this technique is a good approach and is available in most commercial CSM codes.[18–21] This approach also provides a mechanism to transfer the sensitivity data to most commercial analysis codes. Because the grids can be regenerated automatically [see Eq. (2)], the approach avoids the need for grid generation. For simple geometry, the design vectors may be manually generated for each discipline. Because multidisciplinary applications often use dissimilar grids, manual generation of consistent design vectors across all disciplines is difficult.

Domain Element Approach

The domain element approach is based on linking a set of grid points to a macroelement (domain element) that controls the shape of the model. Figure 4a shows a domain element with four nodes (A–D) for the baseline model. As the nodes of the domain element move (A–D), the grid points belonging to the domain will move as well (Fig. 4b). The movement is based on an inverse mapping between the grid points and the domain element, and the parametric coordinates of the grid points with respect to the domain element are kept fixed through the optimization cycles.[19] The domain element technique is available for shape optimization in some commercial software.[21] This method is very efficient, and it is relatively simple to implement. Because the method is based on parameterizing a set of points regardless of their connectivity, it would 1) avoid grid regeneration and 2) result in a consistent shape parameterization for multidisciplinary applications.

Partial Differential Equation Approach

Bloor and Wilson[22] presented an efficient and compact method for parameterizing the surface geometry of an aircraft. The method considers the surface generation as a boundary-value problem, and it produces surfaces as the solutions to elliptic partial differential equations (PDE). Bloor and Wilson showed that it was possible to represent aircraft geometry in terms of a small set of design variables. Brown et al.[23] presented methods for generating B-spline approximations to PDE surfaces to enable data transfer to CAD systems. Smith et al.[24] extended the PDE approach to a class of airplane configurations. Included in this definition were surface grids, volume grids, and grid sensitivity derivatives for CFD. Grid sensitivity was obtained by applying the automatic differentiation tool ADIFOR.[8]

Discrete Approach

The discrete approach is based on using the grid-point coordinates (Fig. 5) as design variables; for example, Refs. 25 and 26. This approach is easy to implement, and the geometry changes are limited only by the number of design variables. Because the geometry is perturbed by moving individual grid points, a smooth geometry is difficult to maintain, and the optimization solution may be impractical to manufacture, as pointed out by Braibant and Fleury.[27] Additional optimization constraints must be taken into account automatically to avoid an unrealistic design. For example, one can use multipoint constraints and dynamic adjustment of lower and upper bounds on the design variables. For a model with a large number of grid points, the number of design variables often becomes very large and may lead to high cost and a difficult optimization problem to solve.

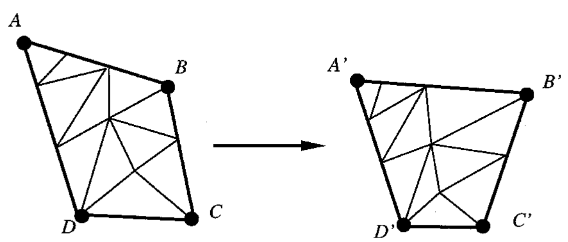
An MDO application often requires parameterization of multiple dissimilar grids, for example, CFD and CSM. Because the discrete approach parameterizes individual grids, it cannot guarantee a consistent shape parameterization across multiple disciplines.

A variation of the discrete approach, the natural design approach, uses a set of fictitious loads as design variables; for example, Ref. 28. These fictitious loads are applied to the boundary points, and the resulting displacements, or natural shape functions, are added to the baseline grid to obtain a new shape. Consequently, the relationship between changes in design variables and grid-point locations is established through a finite element analysis. Zhang and Belegundu[29] provided a systematic approach for generating the sensitivity derivatives and several criteria to determine their effectiveness.

The most attractive feature of the discrete approach is the ability to use an existing grid for optimization, and the grid regeneration process can be avoided during the shape optimization process. The model complexity has little or no bearing on the parameterization process. A strong local control on shape changes can be achieved by restricting the changes to a small area. When the shape design variables are the grid-point coordinates, the grid sensitivity derivative analysis is trivial to calculate; the third and fourth right-hand terms in Eq. (1) can be combined to form an identity matrix.

Polynomial and Spline Approaches

Use of polynomial and spline representations for shape parameterization obviously can reduce the total number of design variables; for example, see Fig. 6. Braibant and Fleury[27] showed that Bezier



a) Baseline model b) Deformed model

Fig. 4 Domain element.

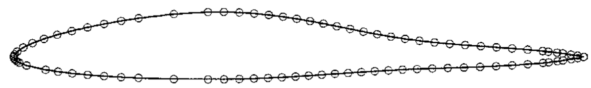


Fig. 5 Airfoil designed by a set of points.

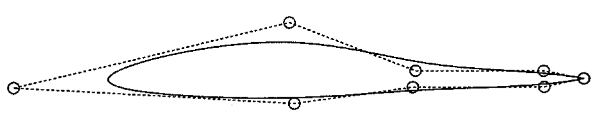


Fig. 6 Airfoil designed by a set of control points.

and B-spline curves are well suited for shape optimization. A polynomial can describe a curve in a very compact form with a small set of design variables.

The analytical sensitivity derivatives with respect to the design variable vector can be computed efficiently and accurately. For example, a curve can be described as the polynomial

$$\bar{\boldsymbol{R}}(u) = \sum_{i=0}^{n-1} \bar{\boldsymbol{c}}_i u^i \tag{3}$$

where n is the number of design variables and u is the parameter coordinate along the curve. The term $\bar{\boldsymbol{c}}_i$ is a set of coefficient vectors corresponding to three-dimensional coordinates, and the components of these vectors can be used as design variables. The sensitivity derivatives of geometry, $\bar{\boldsymbol{R}}$, with respect to $\bar{\boldsymbol{c}}_i$ are u^i. The polynomial representation in Eq. (3) is in the power basis form, and the $\bar{\boldsymbol{c}}_i$ coefficient vectors convey very little geometric insight about the shape. Also, the power basis form is prone to round-off error if there is a large variation in the magnitude of the coefficients.[30] Nevertheless, the polynomial form is a powerful and compact representation for shape optimization of simple curves, for example, Refs. 31 and 32.

The Bezier representation is another mathematical form for representing curves and surfaces. For example, a Bezier curve can be described by

$$\bar{\boldsymbol{R}}(u) = \sum_{i=1}^{n} \bar{\boldsymbol{P}}_i B_{i,p}(u) \tag{4}$$

where n is the number of control points (design variables) and the $B_{i,p}(u)$ are degree p Bernstein polynomials. The coefficients $\bar{\boldsymbol{P}}_i$ are the control points (forming a control polygon), and they typically are used as design variables. See Farin[30] for further discussions on the properties of Bezier form. The Bezier form is a far better representation than the power basis, even though mathematically equivalent, because the computation of Bernstein polynomials is a recursive algorithm (de Casteljau algorithm) (see Ref. 30) that minimizes the roundoff error. Also, the control points are more closely related to the curve position. In fact, the control points approximate the curve. The convex hull of the Bezier control polygon contains the curve. This property is very useful, especially in defining the geometric constraints. The first and last control points are located exactly at the beginning and the end of the curve, respectively. The sensitivity derivatives of geometry, $\bar{\boldsymbol{R}}$, with respect to $\bar{\boldsymbol{P}}_i$ are $B_{i,p}(u)$, the Bernstein polynomial functions. These functions are independent of the Bezier control points, that is, design variables; therefore, the sensitivity derivatives stay fixed during the optimization process.

The Bezier form is an effective and accurate representation for shape optimization of simple curves, for example, Ref. 33. Complex curves require a high-degree Bezier form. However, as the degree of a Bezier curve increases, so does the roundoff error. Also, it is very inefficient to compute a high-degree Bezier curve. Instead, to use Bezier representation for a complex curve, one can use several low-degree Bezier segments to cover the entire curve. The resulting composite curve is referred to as a spline or, more accurately, a B-spline. A multisegmented B-spline curve can be described by

$$\bar{\boldsymbol{R}}(u) = \sum_{i=1}^{n} \bar{\boldsymbol{P}}_i N_{i,p}(u) \tag{5}$$

where $\bar{\boldsymbol{P}}_i$ are the B-spline control points, p is the degree, and $N_{i,p}(u)$ is the ith B-spline basis function of degree p. In addition to the desirable properties of the Bezier representation, the low-degree B-spline form can represent complex curves efficiently and accurately. The sensitivity derivatives of geometry, $\bar{\boldsymbol{R}}$, with respect to $\bar{\boldsymbol{P}}_i$ are $N_{i,p}(u)$, the B-spline basis function. Similar to a Bezier form, the sensitivity derivatives of a B-spline curve stay fixed during the optimization cycles.

Some limited applications in the literature are based on polynomial and spline representations. Cosentino and Holst[15] optimized a transonic wing configuration by using a cubic-spline representation for two-dimensional airfoils that define wing geometry. Then, they used the position of the spline control points, in particular, those points that affect the wing region wetted by supersonic flow, as design variables to be optimized. In a design case study on the Lockheed C-141B aircraft, Cosentino and Holst reduced the number of design variables from 120 to 12 by using the cubic-spline technique. In recent years, Schramm and Pilkey[34] used a B-spline representation to perform structural shape optimization for a torsion problem with direct integration and B-splines. Similarly, Anderson and Venkatakrishnan[35] used B-splines with an unstructured grid CFD code for aerodynamic design optimization.

The only drawback of the regular B-spline representation is its inability to represent implicit conic sections accurately. However, a special form of B-spline, nonuniform rational B-spline (NURBS), can represent most parametric and implicit curves and surfaces without loss of accuracy.[30] NURBS can represent quadric primitives, for example, cylinders and cones, as well as free-form geometry.[30] Some implicit surfaces, for example, helix and helicoidal[36] cannot be converted directly to NURBS, but these surfaces are not common in most aerospace applications. A NURBS curve is defined as

$$\bar{\boldsymbol{R}}(u) = \frac{\sum_{i=1}^{n} N_{i,p}(u) W_i \bar{\boldsymbol{P}}_i}{\sum_{i=1}^{n} N_{i,p}(u) W_i} \tag{6}$$

where the $\bar{\boldsymbol{P}}_i$ are the control points, W_i are the weights, and $N_{i,p}(u)$ is the ith B-spline basis function of degree p. Similar to the Bezier form, for a NURBS representation the sensitivity derivatives with respect to the control points are fixed during the optimization cycles. However, if the weights are selected as design variables, the sensitivity derivatives will be functions of the weight design variables. Schramm et al.[37] have successfully used the two-dimensional NURBS representations for shape optimization.

The polynomial and spline techniques are well suited for two-dimensional and simple three-dimensional models. Complex three-dimensional models are made of many curves and surfaces; as a result, these curves and surfaces are difficult to model outside of a CAD system. Also, complex models require a large number of control points, and optimization is prone to creating irregular[27] or wavy[38] geometry.

CAD-Based Approach

Use of commercial CAD systems for geometry modeling can potentially save development time and be the single source for geometry construction and manipulation for an MDO application. For a more detailed account of the role of CAD in MDO, see Ref. 14. Most solid modeling CAD systems use either a boundary representation or a constructive solid geometry method to represent a physical, solid object.[39]

To parameterize an existing model is still a challenging task in today's CAD systems,[40] and the models created are not always good enough for automatic grid generation tools. Designers may believe their models are complete and accurate, but unseen imperfections (e.g., gaps, unwanted wiggles, free edges, slivers, and transition cracks) often cause problems in gridding for CSM and CFD. For more details, see Refs. 14 and 41.

Feature-based solid modeling (FBSM) CAD systems[42] are capable of creating dimension-driven objects. These systems use Boolean operations such as intersection and union of simple features. Examples of simple features include holes, slots (or cuts), bosses (or protrusions), fillets, chamfers, sweeps, and shells. Today's CAD systems allow designers to work in a three-dimensional space while using topologically complete geometry (solid models) that can be modified by altering the dimensions of the features from which it was created. The most important capability of FBSM is the ability to capture the design intent. The FBSM tools have made design modification much easier and faster. The developers of FBSM CAD systems have put the design back in CAD. Because FBSM CAD tools enable today's design engineers to create a new, complete, and parametric model for a configuration, these tools are being incorporated into the design environment. Blair and Reich[43] presented a vision to integrate an FBSM CAD system with full associativity into a virtual design environment.

Even though use of parametric modeling in design would make the FBSM tools ideal for optimization, existing FBSM tools are not capable of calculating sensitivity derivatives analytically. Townsend et al.[40] discussed issues involved in using a CAD system for an MDO application. This work identified the calculation of the analytical sensitivity derivative as one of the important integration issues. The computer codes for commercial CAD systems are very large; to differentiate the entire system with automatic differentiation tools may not be feasible. Therefore, calculation of the analytical sensitivity derivatives of geometry with respect to the design variables could prove to be difficult within a commercial CAD environment. For some limited cases, the analytical shape sensitivity derivatives can be calculated based on a CAD model[44]; however, this method will not work under all circumstances. One difficulty is that, for some perturbation of some dimensions, the topology of the CAD part may be changed.

Another way to calculate the sensitivity derivatives is to use finite differences, as long as the perturbed geometry has the same topology as the unperturbed geometry. Both methods, the analytical and finite difference approximations, have their difficulties and limitations. He et al.[45] presented a procedure for integrating CAD and computer-aided engineering systems to support geometry- and detailed- analysis-based optimization. The sensitivity derivatives were calculated by a finite difference approximation.

Analytical Approach

Hicks and Henne[16] introduced a compact formulation for parameterization of airfoil sections. The formulation was based on adding shape functions (analytical functions) linearly to the baseline shape. The contribution of each parameter is determined by the value of the participating coefficients (design variables) associated with that function. All participating coefficients are initially set to zero, and so the first computation gives the baseline geometry. The shape functions are smooth functions based on a set of previous airfoil designs. Elliott and Peraire[32] and Hager et al.[46] used a formulation similar to that of Hicks and Henne,[16] but with a different set of shape functions. This method is very effective for wing parameterization.

FFD Approach

Grid generation for CFD and CSM is time consuming and costly for a full airplane model: To develop detailed CSM and CFD grids based on a CAD model requires several months. To fit into the product development cycle times, the MSO must rely on the parameterization of the analysis grids, for which the FFD algorithm is ideal. The FFD algorithm is a subset of the soft object animation (SOA) algorithms used in computer graphics[47] for morphing images[48] and deforming models.[49,50] These algorithms are powerful tools for modifying shapes: They use a high-level shape deformation, as opposed to manipulation of lower level geometric entities. The deformation algorithms are suitable for deforming models represented by either a set of polygons or a set of parametric curves and surfaces. The SOA algorithms treat the model as rubber that can be twisted, bent, tapered, compressed, or expanded, while retaining its topology. This algorithm is ideal for parameterizing airplane models that have external skin as well as internal components; for example, see Fig. 1. The SOA algorithms relate the grid-point coordinates of an analysis model to a number of design variables. Consequently, the SOA algorithms can serve as the basis for an efficient shape parameterization technique.

Barr[49] presented a deformation approach in the context of physically based modeling. This approach uses physical simulation to obtain realistic shape and motions and is based on operations such as translation, rotation, and scaling. With this algorithm, the deformation is achieved by moving the grid points of a polygon model or the control points of a parametric curve and surface. Sederberg and Parry[50] presented another approach for deformation, based on the FFD algorithm, that operates on the whole space regardless of the representation of the deformed objects embedded in the space. The algorithm allows a user to manipulate the control points of trivariate Bezier volumes. Coquillart[51] extended a Bezier parallelepiped to a nonparallelepiped cubic Bezier volume.

Lamousin and Waggenspack[52] modified FFD to include NURBS definition and multiple blocks to model complex shapes. Yeh and Vance[53] and Perry and Balling[54] used the modified technique for design and optimization. Yeh and Vance[53] developed an application based on NURBS whereby the user can change the shape of a virtual object and examine the effect the shape change has on the displacement of the structural deformation and stress distribution throughout the object. Perry et al.[55] successfully used the FFD algorithm for the optimization of an automobile air conditioning duct system.

Hsu et al.[56] presented a method to manipulate directly the object; this method creates a more intuitive and transparent environment for FFD. Borrel and Rappoport[57] presented a simple, constrained deformation that allows the user to define a set of constraint points, giving a desired displacement and radius of influence for each. Each constraint point determines a local B-spline basis function, centered at the constraint point that falls to zero for points beyond the radius. This technique directly influences the final shape of the deformed object.

The FFD formulation is independent of grid topology, and that independence makes it suitable for a variety of analysis codes, such as low-fidelity (e.g., linear aerodynamics and equivalent laminated plate structures) and high-fidelity (e.g., nonlinear CFD and detailed finite element modeling) analysis tools. The analytical sensitivity derivatives are available for use in a gradient-based optimization.

The design variables used in FFD may have no physical significance for the design engineers, thereby making it difficult to establish an effective and compact set of design variables. To resolve this difficulty, the original SOA algorithms have been modified,[58] and the modified algorithms are referred to as multidisciplinary aerodynamic-structural shape optimization using deformation (MASSOUD).

The MASSOUD approach consists of three basic concepts: 1) parameterizing the shape perturbations rather than the geometry itself, 2) utilizing the SOA algorithms used in computer graphics, and 3) relating the deformation to aerodynamic shape design variables such as thickness, camber, twist, shear, and planform. Reference 58 contains the implementation details of parameterizing for planform, twist, dihedral, thickness, and camber.

The modified algorithm has been used for parameterizing a simple wing, a blended wing body, and several high-speed civil transport configurations. The algorithm has been successfully implemented for aerodynamic shape optimization with analytical sensitivity derivatives for structured grid[59] and unstructured grid[60] CFD codes.

Field Grid Regeneration and Deformation

During the shape optimization process, the surface grids are either regenerated or deformed. As a result, the field grids for high-fidelity analysis tools, such as CFD, must be either regenerated or deformed. The next two subsections provide an overview for structured and unstructured grid regeneration and deformation techniques.

Structured Grid

Most structured grid regeneration and deformation techniques are based on transfinite interpolation (TFI). Gaitonde and Fiddes[61] used a regenerating grid technique based on using TFI with exponential blending functions. The choice of blending functions has a considerable influence on the quality and robustness of the field grid. Soni[62] proposed a set of blending functions based on arc length that is extremely effective and robust for grid regeneration and deformation. His algorithm has been incorporated in most commercial structured grid generation packages.

Jones and Samareh[6] presented an algorithm for grid regeneration and deformation based on Soni's[62] blending functions, and they also provided analytical sensitivity derivatives by using the automatic differentiation tool ADIC.[9] The method is suitable for a general, multiblock, three-dimensional volume grid deformation. Hartwich and Agrawal also used the idea of volume grid deformation.[63] They introduced two new techniques: 1) the use of the slave–master concept to semiautomate the process and 2) the use of a Gaussian distribution function to preserve the integrity of grids in the presence

Table 1 Multidisciplinary shape parameterization approaches

Feature	Basis vector	Domain element	PDE	Discrete	Polynomial and spline	CAD based	Analytical	FFD
What parameterizing capabilities?	G[a]	G	S[b]	G	S	S	G, S	G, S
Consistent across disciplines?		Y[c]	Y		Y	Y	Y	Y
Require reverse engineering of original design?			Y		Y	Y		
Require automatic grid generation tools?			Y		Y	Y		
Handle large geometry changes?			Y		Y	Y		
Analytical sensitivity derivatives available?	Y	Y	Y	Y	Y		Y	Y
Level of complexity for geometry changes?	L[d]	L	M[e]	L	M	H[f]	L	M

[a]Grid. [b]Surface. [c]Yes. [d]Low. [e]Medium. [f]High.

of multiple body surfaces. Reuther et al.[64] used a modified TFI approach with blending functions based on arc length, and they used finite difference approximation to compute the sensitivity derivatives for the field grid.

Leatham and Chappell[65] used the Laplacian technique, commonly used for unstructured grid deformation, for moving structured grids. They have been successful in deforming structured grids with this technique.

Unstructured Grid

For unstructured grids with large geometric changes, Botkin[66] proposed to regenerate a complete grid at the beginning of each optimization cycle. However, for gradient calculations many small changes must be made, and to regenerate the grid for each design variable perturbation would be too costly. Botkin has introduced a local regridding procedure that operates only on the specific edges and faces associated with the design variables being perturbed. Similarly, Kodiyalam et al.[67] used a grid regeneration technique based on the assumption that the solid model topology stays fixed for small perturbations. The solid model topology contains the number of grid points, edges, and faces. Any change in the topology will cause the model regeneration to fail. To avoid such a failure, a set of constraints must be satisfied among design variables, in addition to constraints on their bounds.

For a dynamic aeroelastic case with unstructured grids, Batina[68] presented a grid deformation algorithm that models grid edges with springs. The spring stiffness for a given edge j–k is taken to be inversely proportional to the element edge length as

$$k_m = 1/\|\bar{\boldsymbol{r}}_j - \bar{\boldsymbol{r}}_k\| \tag{7}$$

The grid movement is computed through predictor and corrector steps. The predictor step is based on an existing solution from the previous cycle, and the corrector step performs several Jacobi iterations of the static equilibrium equations by using

$$\bar{\boldsymbol{D}}^{n+1} = \frac{\sum k_m \bar{\boldsymbol{D}}^n_m}{\sum k_m} \tag{8}$$

where the summation is over all edges of the elements. This corrector is similar to a Laplace operator, which has a diffusive behavior. In contrast to its use for dynamic aeroelasticity, the previous optimization cycle may not provide a good initial guess for the corrector step.

Zhang and Belegundu[29] proposed a similar algorithm to handle large grid movement. The equation for grid update is similar to Batina's[68] approach:

$$\bar{\boldsymbol{R}}^{\text{new}} = \frac{\sum k_m \bar{\boldsymbol{R}}^{\text{old}}}{\sum k_m}, \quad \text{where} \quad k_m = \frac{8|J|}{V} \tag{9}$$

where J is the cell Jacobian defined within cell parametric coordinates and V is the cell volume.

Crumpton and Giles[69] found the spring analogy to be inadequate and ineffective for large grid perturbations. They proposed a technique based on using the heat transfer equation, where $k_m = 1/\max(V, \varepsilon)$,

$$\nabla \cdot \{k_m \nabla(\bar{\boldsymbol{D}})\} = 0 \tag{10}$$

The term V is the cell volume, and ε is a small positive number needed to avoid a division by zero. This technique is similar to the spring analogy,[68] except that it uses the cell volume for k_m. The coefficient k_m is relatively large for small cells. Therefore, these small cells, which are usually near the surface of the body, tend to undergo rigid-body motion. This rigid-body movement avoids rapid variations in $\bar{\boldsymbol{D}}$, thus eliminating the possibility of small cells having very large changes in volume; such changes could lead to negative cell volumes. Crumpton and Giles[69] used an underrelaxed Jacobi iteration, with the nonlinear k_m evaluated at the previous iteration.

Summary

Eight shape parameterization approaches were presented. The choice of shape parameterization approach depends on 1) consistency and accuracy of the geometry representation, 2) number of disciplines involved, 3) availability of automatic grid generation, 4) optimization algorithms and requirement on availability of analytical sensitivity, 5) development cycle time, and 6) direct connection to CAD. Table 1 summarizes the eight approaches surveyed in this paper.

References

[1]Haftka, R. T., and Grandhi, R. V., "Structural Shape Optimization—A Survey," *Computer Methods in Applied Mechanics and Engineering*, Vol. 57, No. 1, 1986, pp. 91–106.

[2]Ding, Y., "Shape Optimization of Structures: A Literature Survey," *Computers and Structures*, Vol. 24, No. 6, 1986, pp. 985–1004.

[3]Livne, E. (ed.), "Multidisciplinary Design Optimization," *Journal of Aircraft*, Vol. 36, No. 1, 1999.

[4]Yang, R. J., and Botkin, M. E., "Accuracy of the Domain Method for the Material Derivative Approach to Shape Design Sensitivities," *AIAA Journal*, Vol. 25, No. 12, 1987, pp. 1606–1610.

[5]Newman, J. C., Taylor, A. C., Barnwell, R. W., Newman, P. A., and Hou, G. J., "Overview of Sensitivity Analysis and Shape Optimization for Complex Aerodynamic Configurations," *Journal of Aircraft*, Vol. 36, No. 1, 1999, pp. 87–96.

[6]Jones, W. T., and Samareh, J. A., "A Grid Generation System for Multidisciplinary Design Optimization," AIAA Paper 95-1689, June 1995.

[7]Zang, T. A., and Green, L. L., "Multidisciplinary Design Optimization Techniques: Implications and Opportunities for Fluid Dynamics Research," AIAA Paper 99-3798, June 1999.

[8]Bischof, C., Carle, A., Khademi, P., and Mauer, A., "ADIFOR 2.0: Automatic Differentiation of Fortran 77 Programs," *IEEE Computational Science and Engineering*, Vol. 3, No. 3, 1996, pp. 18–32.

[9]Bischof, C., Roh, L., and Mauer, A., "ADIC—an Extensible Automatic Differentiation Tool for ANSI-C," *Software—Practice and Experience*, Vol. 27, No. 12, 1997, pp. 1427–1456.

[10]Squire, W., and Trapp, G., "Using Complex Variables to Estimate Derivatives of Real Functions," *SIAM Review*, Vol. 10, No. 1, 1998, pp. 110–112.

[11]Anderson, K., Nielson, E., Whitfield, D., and Newman, J., "Sensitivity Analysis for the Navier–Stokes Equations on Unstructured Meshes using Complex Variables," AIAA Paper 99-3294, 1999.

[12]Gill, P. E., Murray, W., Saunders, M. A., and Wright, M. H., "Computing Finite-Difference Approximation to Derivatives for Numerical Optimization," Dept. Operations Research—SOL, Rept. SOL 80-6, Stanford Univ., Stanford, CA, May 1980.

[13]Lott, J., Haftka, R. T., and Adelman, H. M., "Selecting Step Sizes in Sensitivity Analysis by Finite Difference," NASA TM-86382, Aug. 1985.

[14]Samareh, J. A., "Status and Future of Geometry Modeling and Grid Generation for Design and Optimization," *Journal of Aircraft*, Vol. 36, No. 1, 1999, pp. 97–104.

[15]Cosentino, G. B., and Holst, T. L., "Numerical Optimization Design of Advanced Transonic Wing Configurations," *Journal of Aircraft*, Vol. 23, No. 3, 1986, pp. 193–199.

[16]Hicks, R. M., and Henne, P. A., "Wing Design by Numerical Optimization," *Journal of Aircraft*, Vol. 15, No. 7, 1978, pp. 407–412.

[17]Pickett, R. M., Rubinstein, M. F., and Nelson, R. B., "Automated Structural Synthesis Using a Reduced Number of Design Coordinates," *AIAA Journal*, Vol. 11, No. 4, 1973, pp. 498–494.

[18]Kodiyalam, S., Vanderplaats, G. N., and Miura, H., "Structural Shape Optimization with MSC/NASTRAN," *Computers and Structures*, Vol. 40, No. 4, 1991, pp. 821–829.

[19]Leiva, J. P., and Watson, B. C., "Automatic Generation of Basis Vectors for Shape Optimization in the GENESIS Program," *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization Conference Proceedings*, AIAA, Reston, VA, 1998, pp. 1115–1122.

[20]Kilroy, K., (ed.), NASTRAN Software User's Manual, MacNeal–Schwendler Corp., Los Angeles, 1998.

[21]GENESIS Structural Optimization Software, Ver. 5.0, VMA Engineering, Colorado Springs, CO, 1997.

[22]Bloor, M. I. G., and Wilson, M. J., "Efficient Parameterization of Generic Aircraft Geometry," *Journal of Aircraft*, Vol. 32, No. 6, 1995, pp. 1269–1275.

[23]Brown, J. M., Bloor, M. I. G., Bloor, M. S., and Wilson, M. J., "The Accuracy of B-Spline Finite Element Approximations to PDE Surfaces," *Computer Methods in Applied Mechanics and Engineering*, Vol. 158, No. 3, 1998, pp. 221–234.

[24]Smith, R. E., Bloor, M. I. G., Wilson, M. J., and Thomas, A. T., "Rapid Airplane Parametric Input Design (RAPID)," *AIAA 12th Computational Fluid Dynamics Conference*, AIAA, Washington, DC, 1995, pp. 452–462.

[25]Campbell, R. L., "An Approach to Constrained Aerodynamic Design With Application to Airfoils," NASA TP-3260, Nov. 1992.

[26]Jameson, A., Pierce, N. A., and Martinelli, L., "Optimum Aerodynamics Design Using the Navier–Stokes Equations," AIAA Paper 97-0101, Jan. 1997.

[27]Braibant, V., and Fleury, C., "Shape Optimal Design Using B-Splines," *Computer Methods in Applied Mechanics and Engineering*, Vol. 44, No. 3, 1984, pp. 247–267.

[28]Belegundu, A. D., and Rajan, S. D., "A Shape Optimization Approach Based on Natural Design Variables and Shape Functions," *Computer Methods in Applied Mechanics and Engineering*, Vol. 66, No. 1, 1988, pp. 87–106.

[29]Zhang, S., and Belegundu, A. D., "A Systematic Approach for Generating Velocity Fields in Shape Optimization," *Structural Optimization*, Vol. 5, No. 1–2, 1993, pp. 84–94.

[30]Farin, G., *Curves and Surfaces for Computer Aided Geometric Design*, Academic Press, New York, 1990.

[31]Taylor, C. A., Hou, G. J., and Korivi, V. M., "Sensitivity Analysis, Approximate Analysis, and Design Optimization for Internal and External Flows," AIAA Paper 91-3083, Sept. 1991.

[32]Elliott, J., and Peraire, J., "Practical Three-Dimensional Aerodynamic Design and Optimization Using Unstructured Meshes," *AIAA Journal*, Vol. 35, No. 9, 1997, pp. 1479–1486.

[33]Baysal, O., and Ghayour, K., "Continuous Adjoint Sensitivities for General Cost Functions on Unstructured Meshes in Aerodynamic Shape Optimization," *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization Conference Proceedings*, AIAA, Reston, VA, Sept. 1998, pp. 1483–1491.

[34]Schramm, U., and Pilkey, W. D., "Structural Shape Optimization for the Torsion Problem Using Direct Integration and B-Splines," *Computer Methods in Applied Mechanics and Engineering*, Vol. 107, Nos. 1–2, 1993, pp. 251–268.

[35]Anderson, W. K., and Venkatakrishnan, V., "Aerodynamics Design Optimization on Unstructured Grids with a Continuous Adjoint Formulation," AIAA Paper 97-0643, Jan. 1997.

[36]Letcher, J. S., and Shook, M., "NURBS Considered Harmful for Gridding (Alternative Offered)," *4th International Meshing Roundtable*, Sandia National Lab., Albuquerque, NM, 1995, pp. 253–264.

[37]Schramm, U., Pilkey, W. D., DeVries, R. I., and Zebrowski, M. P., "Shape Design for Thin-Walled Beam Cross Sections Using Rational B-Splines," *AIAA Journal*, Vol. 33, No. 11, 1995, pp. 2205–2211.

[38]Jameson, A., and Reuther, J., "A Comparison of Design Variables for Control Theory Based Airfoil Optimization," *6th International Symposium on Computational Fluid Dynamics*, Vol. 4, Lake Tahoe, NV, 1995, pp. 101–107.

[39]LaCourse, D. E., *Handbook of Solid Modeling*, McGraw–Hill, New York, 1995.

[40]Townsend, J. C., Samareh, J. A., Weston, R. P., and Zorumski, W. E., "Integration of a CAD System Into an MDO Framework," NASA TM-207672, May 1998.

[41]Ferguson, D. R., Lucian, M. L., and Seitelman, L., "PDES, Inc., Geometric Accuracy Team Interim Report," Boeing Information and Support Services, Rept. ISSTECH-96-013, The Boeing Co. Seattle, WA, Nov. 1996.

[42]Shah, J. J., and Mantyla, M., *Parametric and Feature-Based CAD/CAM*, Wiley, New York, 1995.

[43]Blair, M., and Reich, G., "A Demonstration of CAD/CAM/CAE in a Fully Associative Aerospace Design Environment," AIAA Paper 96-1630, April 1996.

[44]Hardee, E., Chang, K.-H., Choi, K. K., Yu, X., and Grindeanu, I., "A CAD-Based Design Sensitivity Analysis and Optimization for Structural Shape Optimization Design Applications," *6th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization Conference Proceedings*, AIAA, Reston, VA, 1996, pp. 77–87.

[45]He, B., Rohl, P. J., Irani, R. K., Thamboo, S. V., and Srivatsa, S. K., "CAD and CAE Integration With Application to the Forging Shape Optimization of Turbine Disks," AIAA Paper 98-2032, April 1998.

[46]Hager, J. O., Eyi, S., and Lee, K. D., "A Multi-Point Optimization for Transonic Airfoil Design," *Proceedings of the 4th AIAA/USAF/NASA/OAI Symposium on Multidisciplinary Analysis and Optimization*, AIAA, Reston, VA, 1992, pp. 13–20.

[47]Watt, A., and Watt, M., *Advanced Animation and Rendering Techniques*, Addison–Wesley, New York, 1992.

[48]Hall, V., "Morphing in 2-D and 3-D," *Dr. Dobb's Journal*, Vol. 18, No. 7, 1993, pp. 18–26.

[49]Barr, A. H., "Global and Local Deformation of Solid Primitives," *Computer Graphics*, Vol. 18, No. 3, 1984, pp. 21–30.

[50]Sederberg, T. W., and Parry, S. R., "Free-Form Deformation of Solid Geometric Models," *Computer Graphics*, Vol. 20, No. 4, 1986, pp. 151–160.

[51]Coquillart, S., "Extended Free-Form Deformation: A Sculpturing Tool for 3D Geometric Modeling," *Computer Graphics*, Vol. 24, No. 4, 1990, pp. 187–196.

[52]Lamousin, H. J., and Waggenspack, W. N., "NURBS-Based Free-Form Deformation," *IEEE Computer Graphics and Applications*, Vol. 14, No. 6, 1994, pp. 59–65.

[53]Yeh, T.-P., and Vance, J. M., "Applying Virtual Reality Techniques to Sensitivity-Based Structural Shape Design," *Proceedings of 1997 ASME Design Engineering Technical Conference*, ASME Vol. DETC97, American Society of Mechanical Engineers, Fairfield, NJ, 1997, pp. 1–9.

[54]Perry, E., and Balling, R., "A New Morphing Method for Shape Optimization," AIAA Paper 98-2896, June 1998.

[55]Perry, E., Balling, R., and Landon, M., "A New Morphing Method for Shape Optimization," AIAA Paper 98-4907, Sept. 1998.

[56]Hsu, W. M., Hughes, J. F., and Kaufman, H., "Direct Manipulation of Free-Form Deformation," *Computer Graphics*, Vol. 26, No. 2, 1992, pp. 177–184.

[57]Borrel, P., and Rappoport, A., "Simple Constrained Deformations for Geometric Modeling and Interactive Design," *ACM Transactions on Graphics*, Vol. 13, No. 2, 1994, pp. 137–155.

[58]Samareh, J. A., "Multidisciplinary Aerodynamic-Structural Shape Optimization Using Deformation (MASSOUD)," AIAA Paper 2000-4911, Sept. 2000.

[59]Biedron, R. T., Samareh, J. A., and Green, L. L., "Parallel Computation of Sensitivity Derivatives With Application to Aerodynamic Optimization of a Wing," *1998 Computational Aerosciences Workshop*, NASA CP-20857, Aug. 1999, pp. 219–224.

[60]Nielsen, E. J., and Anderson, W. K., "Aerodynamic Design Optimization on Unstructured Meshes Using the Navier–Stokes Equations," *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization Conference Proceedings*, AIAA, Reston, VA, 1998, pp. 825–837.

[61]Gaitonde, A. L., and Fiddes, S. P., "A Three-Dimensional Moving Mesh Method for the Calculation of Unsteady Transonic Flows," *Aeronautical Journal*, Vol. 99, No. 984, 1995, pp. 150–160.

[62]Soni, B. K., "Two- and Three-Dimensional Grid Generation for Internal Flow Applications," AIAA Paper 85-1526, 1985.

[63]Hartwich, P. M., and Agrawal, S., "Method for Perturbing Multiblock Patched Grids in Aeroelastic and Design Optimization Applications," AIAA Paper 97-2038, 1997.

[64]Reuther, J. J., Jameson, A., Alonso, J. J., Rimlinger, M. J., and Saunders, D., "Constrained Multipoint Aerodynamic Shape Optimization Using an Adjoint Formulation and Parallel Computers, Part 1," *Journal of Aircraft*, Vol. 36, No. 1, 1999, pp. 51–60.

[65]Leatham, M., and Chappell, J. A., "On the Rapid Regeneration of Hybrid Grids Due to Design Driven Geometry Perturbation," *The Sixth International Conference on Numerical Grid Generation in Computational Field Simulations*, Mississippi State Univ., Mississippi State, MS, 1998, pp. 533–542.

[66]Botkin, M. E., "Three-Dimensional Shape Optimization Using Fully Automatic Mesh Generation," *AIAA Journal*, Vol. 30, No. 5, 1992, pp. 1932–1934.

[67]Kodiyalam, S., Kumar, V., and Finnigan, P., "Constructive Solid Geometry Approach to Three-Dimensional Structural Shape Optimization," *AIAA Journal*, Vol. 30, No. 5, 1992, pp. 1408–1415.

[68]Batina, J. T., "Unsteady Euler Airfoil Solutions Using Unstructured Dynamic Meshes," *AIAA Journal*, Vol. 28, No. 8, 1990, pp. 1381–1388.

[69]Crumpton, P. I., and Giles, M. B., "Implicit Time-Accurate Solutions on Unstructured Dynamic Grids," *International Journal for Numerical Methods in Fluids*, Vol. 25, No. 11, 1997, pp. 1285–1300.

J. Kallinderis
Associate Editor